PUC Minas

PUC Minas Praça da Liberdade
Curso de Bacharelado em Engenharia de Software
Projeto e Análise de Algoritmos
Laboratório de Computação III - Semestre 1/2020
Prof. João Caram
Trabalhos práticos - Valor: 25 pontos

Regras:
- Trabalho em grupo de até 4 alunos
- Cópias receberão nota 0, sem direito a recurso
- Haverá apresentação dos resultados do trabalho, com a especificação individual de tarefas e a nota podendo ser atribuída individualmente a cada membro do grupo.

Na última unidade de ensino das nossas disciplinas, estudamos diversas técnicas de projetos de algoritmos como alternativas para a tentativa da obtenção da solução ótima em problemas de diferentes áreas. Para este trabalho final, que vale pontos tanto para PAA como para Laboratório de Computação III, seu grupo de trabalho deve:

a) Implementar, obrigatoriamente, o problema P1;

b) Dentre os demais problemas (P2 a P7), escolher UM deles e implementar sua solução com força bruta e mais uma técnica à sua escolha. Neste caso, deve ser investigado (a) o tempo necessário para resolver instâncias crescentes do problema proposto com as duas técnicas e (b) o tamanho do maior conjunto possível com solução obtida por força bruta no tempo-limite de 5 segundos.

c) Dentre os problemas não escolhidos na letra (b), escolher DOIS deles e implementar sua solução com uma técnica que resulte em um algoritmo eficiente.

d) Formar o grupo de trabalho e preencher a planilha disponibilizada pelo professor, indicando membros do grupo e atribuições de cada um no trabalho.

Obs.: a análise do problema e a pesquisa de alternativas para sua solução fazem parte das atribuições do grupo para resolver o trabalho. Naturalmente, o grupo pode – e deve – se dirigir aos professores para debater ideias e algoritmos para os problemas escolhidos.

P1 – Problema da Mochila

Dada uma mochila de capacidade máxima C e um conjunto de N itens, cada um com um valor V e um peso P, encontrar o subconjunto de itens que possa ser levado na mochila de modo a maximizar o valor transportado, sem exceder a capacidade permitida.

As soluções devem ser apresentadas utilizando *força bruta, algoritmo guloso* e *programação dinâmica*, conforme estudado nas aulas. As soluções devem ser comparadas em termos de precisão e tempo de execução.

Para os testes, o grupo pode usar o gerador de conjuntos aleatórios fornecidos no laboratório.

P2 – Par de pontos mais próximos

Dado um conjunto de N pontos em um plano cartesiano, representados por um par de coordenadas *(x,y)*, dizer qual par de pontos apresenta a menor distância euclidiana entre si.

O grupo pode usar o gerador de pontos aleatórios fornecidos no laboratório, o qual usa a classe *Point* do Java.

P3 – Lista de supermercado

Dada uma lista de N produtos identificados por um código, seu preço e seu peso, montar uma lista de compras de supermercado que inclua a maior quantidade destes produtos sem ultrapassar um orçamento O nem um peso máximo P.

O grupo pode usar o gerador de produtos aleatórios fornecidos no laboratório.

P4 – Robô de limpeza

Dado um mapa de N x M quadrículas representando uma residência e uma quadrícula de saída S, encontrar o caminho que um robô de limpeza deve percorrer para limpar toda a residência. O mapa pode conter obstáculos, os quais o robô não consegue ultrapassar. Pode haver mais de um caminho possível e valorizam-se soluções eficientemente calculadas, ainda que o caminho não seja mínimo.

O grupo pode gerar mapas aleatórios com 0 representando uma quadrícula livre e 1 representando obstáculos – mas fiquem atentos a mapas que criem zonas isoladas e inacessíveis a partir de S.

P5 – Agendamento de tarefas em um auditório

Seja $X = \{x_1, x_2, ..., x_n\}$ um conjunto de atividades que estão planejadas para acontecer em um auditório – tão logo as condições e autoridades sanitárias permitam!!! Este auditório tem um horário de abertura S e um horário de fechamento C. Cada atividade x_i possui o tempo t_i necessário para ser completada e uma hora h_i a partir da qual pode ser iniciada.

Selecione o maior subconjunto de tarefas $A \subset X$ nestas condições, ou seja: crie um agendamento para o maior número possível de atividades sem que uma se sobreponha à outras e sem estourar a hora de fechamento do auditório.

O grupo pode usar o gerador de conjuntos aleatórios fornecidos no laboratório.

P6 – Coincidência máxima de subsequência

Sejam S_1 e S_2 duas *strings* de tamanho arbitrário. Uma *subsequência* de letras nestas *strings* é um conjunto de letras $L[i_1]$, $L[i_2]$, $L[i_3]$ … $L[i_n]$ em S_1 que encontra correspondência exata em S_2, considerando $i_1 < i_2 < i_3 < ... < i_n$.

Procure, para estas *strings*, qual é a maior *subsequência* comum às duas. Veja o exemplo:

S_1 = NOTURNO

S_1 = MOSQUITEIRO

a maior subsequência comum é **O U R O:**

S_1 = N**O**T**UR**N**O** (posições 1, 3, 4, 6)

S_1 = M**O**SQ**U**ITEI**RO** (posições 1, 4, 9, 10)

Este problema encontra aplicações em áreas tão variadas como análise de textos, reconhecimento de padrões, comparação de arquivos, sequenciamento de genomas, entre outras.

P7 – Torres de celular

Considere um mapa quadriculado M de dimensões N x N. Considere um conjunto de torres de transmissão $X = \{T_1, T_2, ..., T_n\}$. Cada torre T_i tem uma capacidade de propagação C, sendo C um número inteiro representando quadrículas nas direções horizontal e vertical.

Crie um algoritmo que seja capaz de dizer se é possível alocar todo o conjunto X em M, sem que a área de propagação de uma torre se sobreponha à de outra. O algoritmo deve informar, do total de N^2, quantas estão cobertas pelas torres. No exemplo abaixo, temos um mapa 6 x 6 abrigando 6 torres com capacidade de propagação 2. São cobertas 31 das 36 casas do mapa.